**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| (51) International Patent Classification 7 : H04L 29/00 | A2 | (11) International Publication Number: WO 00/14938 |
|---|---|---|
| | | (43) International Publication Date: 16 March 2000 (16.03.00) |

(21) International Application Number: PCT/US99/20158

(22) International Filing Date: 1 September 1999 (01.09.99)

(30) Priority Data:
09/150,630 9 September 1998 (09.09.98) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, M/S PAL01-521, Palo Alto, CA 94303 (US).

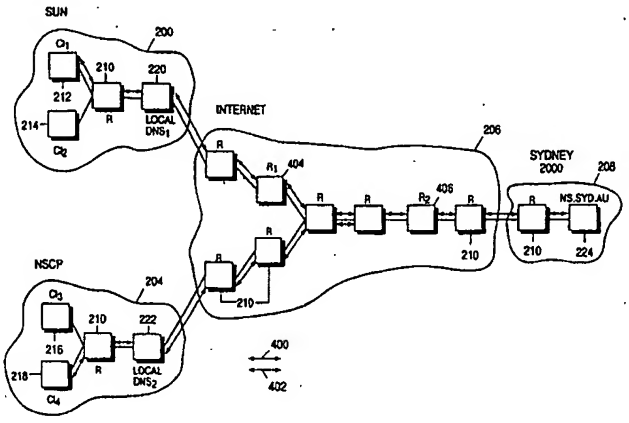
(72) Inventors: GUPTA, Amit; Apartment J207, 2000 Walnut Avenue, Fremont, CA 94538 (US). SCHUBA, Christoph; 473 Hope Street #1, Mountain View, CA 94041 (US). BAEHR, Geoffrey; 531 Colorado AVenue, Palo Alto, CA 94306 (US).

(74) Agents: HECKER, Gary, A. et al.; Hecker & Harriman, Suite 2300, 1925 Century Park East, Los Angeles, CA 90067 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**
*Without international search report and to be republished upon receipt of that report.*

(54) Title: METHOD AND APPARATUS FOR TRANSPARENTLY PROCESSING DNS TRAFFIC




(57) Abstract

A method and apparatus for transparently processing DNS traffic. To access information on the internet using a domain name, the internet protocol (IP) address that maps to the host name must be determined. The host name system (DNS) is utilized to transmit and process the address and domain name information. DNS traffic comprises approximately 10 % of the internet network traffic. When a client requests a name server to translate a domain name into an IP address, the requests are forwarded from one network router to another network router until a name server that maintains the desired information is reached. The network routers do not examine the information, but merely forward the information along the pathway to the destination name server. One or more embodiments of the invention provide for updated routers that recognize when the information consists of DNS traffic, parses the information, caches the address information (if any), and then continues to forward the desired information back to the client of the name service. Consequently, when another request for similar address information is forwarded to a router, the router can provide the response to the requestor instead of forwarding the request to a distant name server. In this manner, routers intercept DNS traffic and cache DNS information, allowing clients that utilize different name servers to benefit from the cached information. Such updated routers reduce the latency in DNS responses and reduce network traffic.

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| AL | Albania | ES | Spain | LS | Lesotho | SI | Slovenia |
| AM | Armenia | FI | Finland | LT | Lithuania | SK | Slovakia |
| AT | Austria | FR | France | LU | Luxembourg | SN | Senegal |
| AU | Australia | GA | Gabon | LV | Latvia | SZ | Swaziland |
| AZ | Azerbaijan | GB | United Kingdom | MC | Monaco | TD | Chad |
| BA | Bosnia and Herzegovina | GE | Georgia | MD | Republic of Moldova | TG | Togo |
| BB | Barbados | GH | Ghana | MG | Madagascar | TJ | Tajikistan |
| BE | Belgium | GN | Guinea | MK | The former Yugoslav Republic of Macedonia | TM | Turkmenistan |
| BF | Burkina Faso | GR | Greece | | | TR | Turkey |
| BG | Bulgaria | HU | Hungary | ML | Mali | TT | Trinidad and Tobago |
| BJ | Benin | IE | Ireland | MN | Mongolia | UA | Ukraine |
| BR | Brazil | IL | Israel | MR | Mauritania | UG | Uganda |
| BY | Belarus | IS | Iceland | MW | Malawi | US | United States of America |
| CA | Canada | IT | Italy | MX | Mexico | UZ | Uzbekistan |
| CF | Central African Republic | JP | Japan | NE | Niger | VN | Viet Nam |
| CG | Congo | KE | Kenya | NL | Netherlands | YU | Yugoslavia |
| CH | Switzerland | KG | Kyrgyzstan | NO | Norway | ZW | Zimbabwe |
| CI | Côte d'Ivoire | KP | Democratic People's Republic of Korea | NZ | New Zealand | | |
| CM | Cameroon | | | PL | Poland | | |
| CN | China | KR | Republic of Korea | PT | Portugal | | |
| CU | Cuba | KZ | Kazakstan | RO | Romania | | |
| CZ | Czech Republic | LC | Saint Lucia | RU | Russian Federation | | |
| DE | Germany | LI | Liechtenstein | SD | Sudan | | |
| DK | Denmark | LK | Sri Lanka | SE | Sweden | | |
| EE | Estonia | LR | Liberia | SG | Singapore | | |

# METHOD AND APPARATUS FOR TRANSPARENTLY PROCESSING DNS TRAFFIC

## BACKGROUND OF THE INVENTION

### 1. FIELD OF THE INVENTION

This invention relates to the field of computer software, and, more specifically, to caching DNS information.



### 2. BACKGROUND ART

In a computer network environment and the internet, computers on the network (clients or servers) are assigned unique identifiers that may be mapped to a textual name referred to as a domain name. Computer users often only have knowledge of the domain name and not the unique identifier. To communicate with a computer on the network, the unique identifier of the computer you are contacting must be ascertained. To

ascertain the unique identifier, network routers forward the identifier request to other routers until a domain name server that maintains the desired information is located. Existing schemes can waste time forwarding the identifier request from one router to another router resulting in an increase of traffic on the network and slowing down the time it takes to access and retrieve any information on the internet. These problems can be understood by reviewing networks, internets, and how they work.

Networks

In modern computing environments, it is commonplace to employ multiple computers or workstations linked together in a network to communicate between, and share data with, network users. A network also may include resources, such as printers, modems, file servers, etc., and may also include services, such as electronic mail.

A network can be a small system that is physically connected by cables (a local area network or "LAN"), or several separate networks can be connected together to form a larger network (a wide area network or "WAN"). Other types of networks include the internet, tel-com networks, the World Wide Web, intranets, extranets, wireless networks, and other networks over which electronic, digital, and/or analog data may be communicated.

Computer systems sometimes rely on a server computer system to provide information to requesting computers on a network. When there are a large number of requesting computers, it may be necessary to have more than one server computer system to handle the requests. In prior art systems,

there is a problem in efficiently directing requests to the correct server in a multiple server system.

One area where this has been a problem is on the internet. The problem can be better understood by reviewing the structure and operation of the internet below.

The Internet

The Internet is a worldwide network of interconnected computers. An Internet client accesses a computer on the network via an Internet provider. An Internet provider is an organization that provides a client (e.g., an individual or other organization) with access to the Internet (via analog telephone line or Integrated Services Digital Network line, for example). A client can, for example, read information from, download a file from or send an electronic mail message to another computer/client using the Internet.

To retrieve a file or service on the Internet, a client must search for the file or service, make a connection to the computer on which the file or service is stored, and download the file or service. Each of these steps may involve a separate application and access to multiple, dissimilar computer systems. The World Wide Web (WWW) was developed to provide a simpler, more uniform means for accessing information on the Internet.

The components of the WWW include browser software, network links, servers. and WWW protocols. The browser software, or browser, is a user-friendly interface (i.e., front-end) that simplifies access to the Internet. A

browser allows a client to communicate a request without having to learn a complicated command syntax, for example. A browser typically provides a graphical user interface (GUI) for displaying information and receiving input. Examples of browsers currently available include Mosaic, Netscape Navigator and Communicator, Microsoft Internet Explorer, and Cello.

Information servers maintain the information on the WWW and are capable of processing a client request. Hypertext Transport Protocol (HTTP) is the standard protocol for communication with an information server on the WWW. HTTP has communication methods that allow clients to request data from a server and send information to the server.

To submit a request, the client contacts the HTTP server and transmits the request to the HTTP server. The request contains the communication method requested for the transaction (e.g., GET an object from the server or POST data to an object on the server). The HTTP server responds to the client by sending a status of the request and the requested information. The connection is then terminated between the client and the HTTP server.

A client request therefore, consists of establishing a connection between the client and the HTTP server, performing the request, and terminating the connection. The HTTP server does not retain any information about the request after the connection has been terminated. HTTP is, therefore, a stateless protocol. That is, a client can make several requests of an HTTP server, but each individual request is treated independent of any other request. The server has no recollection of any previous request.

An addressing scheme is employed to identify Internet resources (e.g., HTTP server, file or program). This addressing scheme is called Uniform Resource Locator (URL). A URL contains the protocol to use when accessing the server (e.g., HTTP), the Internet domain name of the site on which the server is running, the port number of the server, and the location of the resource in the file structure of the server.

The WWW uses a concept known as hypertext. Hypertext provides the ability to create links within a document to move directly to other information. To activate the link, it is only necessary to click on the hypertext link (e.g., a word or phrase). The hypertext link can be to information stored on a different site than the one that supplied the current information. A URL is associated with the link to identify the location of the additional information. When the link is activated, the client's browser uses the link to access the data at the site specified in the URL.

If the client request is for a file, the HTTP server locates the file and sends it to the client. An HTTP server also has the ability to delegate work to gateway programs. The Common Gateway Interface (CGI) specification defines a mechanism by which HTTP servers communicate with gateway programs. A gateway program is referenced using a URL. The HTTP server activates the program specified in the URL and uses CGI mechanisms to pass program data sent by the client to the gateway program. Data is passed from the server to the gateway program via command-line arguments, standard input, or environment variables. The gateway program processes the data and returns its response to the server using CGI (via standard input, for example). The server forwards the data to the client using the HTTP.

A browser displays information to a client/user as pages or documents (referred to as "web pages" or "web sites"). A language is used to define the format for a page to be displayed in the WWW. The language is called Hypertext Markup Language (HTML). A WWW page is transmitted to a client as an HTML document. The browser executing at the client parses the document and displays a page based on the information in the HTML document.

HTML is a structural language that is comprised of HTML elements that are nested within each other. An HTML document is a text file in which certain strings of characters, called tags, mark regions of the document and assign special meaning to them. These regions are called HTML elements. Each element has a name, or tag. An element can have attributes that specify properties of the element. Blocks or components include unordered list, text boxes, check boxes, radio buttons, for example. Each block has properties such as name, type, and value. The following provides an example of the structure of an HTML document:

```
<HTML>
    <HEAD>
    .... element(s) valid in the document head
    </HEAD>
    <BODY>
    .... element(s) valid in the document body
    </BODY>
</HTML>
```

Each HTML element is delimited by the pair of characters "<" and ">". The name of the HTML element is contained within the delimiting characters. The combination of the name and delimiting characters is referred to as a marker, or tag. Each element is identified by its marker. In most cases, each element has a start and ending marker. The ending marker

is identified by the inclusion of an another character, "/" that follows the "<" character.

HTML is a hierarchical language. With the exception of the HTML element, all other elements are contained within another element. The HTML element encompasses the entire document. It identifies the enclosed text as an HTML document. The HEAD element is contained within the HTML element and includes information about the HTML document. The BODY element is contained within the HTML. The BODY element contains all of the text and other information to be displayed. Other HTML elements are described in HTML reference manuals.

Domain Name Server

A computer user navigates the internet or web from a browser on a computer system. To access a web site, the user enters the host name (or domain name) of the web site into the browser. This can be accomplished by clicking on a link, by activating a tool bar button, or by manually entering a name or address into a location field and pressing "enter". The names that a browser client uses are known as host names, such as www.sun.com for example. The name that is entered is not the actual Internet Protocol (IP) address of the intended web server. The actual IP address is a string of numbers that uniquely locate the web server that provides the web site data. A worldwide distributed database system, called the "Domain Name System (DNS)" provides the mapping between server names and the associated IP addresses.

Each client (or host) is configured with, or otherwise learns about, a name server that is willing to answer its queries (for mapping a domain name to an IP address, or vice versa). Such a name server is referred to as the "local name server" for that host. Client application software, such as a web browser, also use a local library, called the "DNS resolver" to obtain the translation from server name to IP address. The resolver in turn contacts a predetermined local DNS name server to obtain the translation. DNS name servers can maintain caches of previously resolved names. More specifically, name resolution processes typically require two hosts on the client side. Consider a user working on "asha.eng.sun.com" that wants to get the address of "whitehouse.gov". The client browser will talk with a local resolver (a library attached to the browser process itself, in the current example running on asha.eng.sun.com). The local resolver will go to one of a relatively small number of local name servers, e.g. "ns.sun.com". Here ns.sun.com is called the client side name server. The client side name server will communicate with the outside world to determine the IP address of whitehouse.gov, and forward this information to the resolver that is part of the browser process.

DNS is a global network of servers that translate host names into numerical addresses (known as Internet Protocol, or IP addresses) and provides IP address to name mapping as well. A DNS server consists of a name server and a resolver. The name server provides responses to resolver requests when it can by supplying the correct address for the host name supplied by the resolver. Referring to Figure 1, at step 100, the user enters the domain name into the browser. At step 102, the browser requests the DNS Resolver to translate the domain name into the IP address. At step, 104, the resolver searches its cache to see if it already has a valid (unexpired) mapping available. If the cache has a valid mapping, it returns the IP address to the

browser at step 116. If the mapping is not in cache, the resolver forwards the request to the local name server at step 106.

All name servers know about at least one other name server that provides the DNS service for the root (.) domain. Thus, at step 108, the local name server contacts the name server for the any known domain. For example, if the host name is "www.java.sun.com", and the local name server does not know the address for the name server "java.sun.com", it will check to see if it knows the next level domain, i.e., the address for "sun.com". If the local name server does not know the address for "sun.com", it will check to see if it knows the address of next level domain, i.e., ".com". If the local name server does not know the address for ".com", it will contact the root name server ".". At step 110, the local name server will obtain the address for the complete domain from the name server contacted (if that name server knows the address). Otherwise, at step 110, the local name server will obtain the address for the next level of the domain from the contacted name server. For example, if the local name server contacted the name server for ".com" and that name server does not know the full address, the ".com" name server will return the domain address for "sun.com". Steps 108 and 110 are then repeated until the complete domain address is obtained. Continuing with the above example, the local name server would contact the "java.com" name server and obtain the address for "java.sun.com". The local name server would then contact the name server for "java.sun.com" and obtain the address for "www.java.sun.com". When a request is made to a name server, there are often many network routers ("routers") that forward the request from one location to another until it reaches the desired name server.

Once an intermediate or complete IP address is obtained, the address is saved in cache so that a future request may be serviced entirely from local cache at step 114. Thus, if a request for an alternative domain is received (e.g., a request for "ftp.sun.com"), the local name server can contact the name server (e.g., "sun.com") directly, without repeating the communication with the root domain server or with intermediate name servers (e.g., the ".com" name server). At step, 116, the IP address is returned to the browser. Once the IP address is known, the browser communicates with the web server at that address to retrieve the requested web page or other information.

The operation of the DNS network is described in:

P.V. Mockapetris "Domain names - concepts and facilities", RFC 1034. Nov 1987.

P.V. Mockapetris "Domain names - implementation and specification", RFC 1035. Nov 1987.

DNS Server Problems

When DNS information is cached in a local name server, the cached information is only available to the clients that access that particular local name server (e.g., clients of the same internet service provider, or members of the same organization). Thus, if two users access different local name servers and each user requests the same IP address, both requests will have to go up the chain of name servers through the various routers, to obtain the needed information.

For example, if two users in different universities in New Zealand were to query the DNS for the IP address of www.sun.com, both of the requests would be serviced by the local name server at ns.sun.com in the United States without any local caching benefit. Figure 2 provides another example of the prior art. Clients $Cl_1$ 212 and $Cl_2$ 214 are part of the SUN network 200 that utilizes local name server $DNS_1$ 220. Clients $Cl_3$ 216 and $Cl_4$ 218 are part on the NSCP network 204 that utilizes local name server $DNS_2$ 222. If client $Cl_1$ 212 requests information regarding an IP address on the SYDNEY 2000 network 208 in Sydney, Australia, the request is processed at the SYDNEY 2000 208 network's local name server ns.syd.au 224. Routers 210 would forward the request from $Cl_1$ to the local name servers 220 that forwards the request through routers 210 on the internet 206 until it reaches the SYDNEY 2000 network 208 and name server 224. The request is then transmitted back along the same route through routers 210 until it returns back to local name server 220 where it is cached.

Only clients that access that same local DNS name server benefit from the caching information. Thus, in the above example, only $Cl_2$ benefits from the $Cl_1$ request and its resulting cached information. If $Cl_4$ requests a DNS translation for www.syd.au, it does not benefit from the cached information, and the information is requested and transmitted all the way to Australia and back. Thus, both $DNS_1$ and $DNS_2$ would obtain the relevant information from Australia creating traffic on the individual networks 200 204 and 208 and internet 206.

Networks may be divided up into layers. For example, one layer may provide for the forwarding of information from one location to another, referred to as the network layer, and another layer may provide for the

parsing and processing of the information passed across the network, referred to as the application layer. Name resolution as provided by the domain name system (DNS) is an application layer protocol. Network routers 210 are only concerned with the network layer protocol and forward the DNS request to its desired destination. Consequently, routers 210 don't parse or process the information that they forward in packets.

Network Traffic Reduction

Prior art methods for reducing network traffic have provided methods for caching web pages and HTML information. Two such prior art methods are referred to as Active Networks and Transparent Proxies.

Active Networks

Routers are dedicated machines for forwarding and switching traffic as quickly as possible. In an Active Network, specific routers are configured to process packets of web and other non-DNS information. Specific geographic locations are chosen to place the specially configured routers. Consequently, the performance of an Active Network is based on the placement strategy of the updated routers.

Transparent Proxies

Transparent Proxies are used mostly by large corporations and internet service providers for reducing their web traffic. In a typical set-up, the domain administrator configures the routers so that all of the web requests (identified by a port number, e.g., 80) are automatically diverted to a proxy server ("transparent proxy"). A proxy server (or proxy) is a server that carries out requests transmitted to it (i.e., from a client), keeping copies of fetched documents or information for some time so that they can be accessed more quickly in the future, speeding up access for commonly requested information. This storing and retrieval of information and fetched documents by the proxy is referred to as caching and the information maintained in the proxy is referred to as a cache or proxy cache. If the proxy does not have the desired information, the proxy sends a request to the appropriate web server (which may be processed through several routers) that then returns the information to the proxy for caching. When the proxy gets the desired information, it provides this information to the requesting client.

The prior art methods do not provide any method for optimizing DNS traffic. Approximately 10% of the traffic on the internet is currently comprised of DNS traffic. Further, since DNS information does not change often (IP addresses often remain the same even when computers on a network are moved), the validity of a DNS entry may be much longer than that of data transmitted through the web. Consequently, an efficient method for optimizing and processing DNS traffic is needed.

## SUMMARY OF THE INVENTION

A method and apparatus for transparently processing DNS traffic. To access information on the internet using a domain name, the internet protocol (IP) address that maps to the domain name must be determined. The domain name system (DNS) is utilized to transmit and process the address and domain name information. DNS traffic comprises approximately 10% of the internet network traffic.

When a client requests a name server to translate a domain name into an IP address, the requests are forwarded from one network router to another network router until a name server that maintains the desired information is located. The network routers do not examine the information, but merely forward the information along the pathway to the destination name server.

One or more embodiments of the invention provide for updated routers that recognize when the information consists of DNS traffic, parses the information, caches the address information (if any), and then continues to forward the desired information back to the name server. Consequently, when another request for similar address information is forwarded to a router, the router can provide the response to the requestor instead of forwarding the request to a distant name server. In this manner, routers intercept DNS traffic and cache DNS information, allowing clients that utilize different name servers to benefit from the cached information. Such updated routers reduce the latency in DNS responses and reduce network traffic.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a prior art method for processing DNS information.

Figure 2 demonstrates the relationship between several networks.

Figure 3 is a block diagram of one embodiment of a computer system capable of providing a suitable execution environment for one or more embodiments of the invention.

Figure 4 demonstrates the relationship between several networks and the path of DNS traffic according to one or more embodiments of the invention.

Figure 5 illustrates the steps executed by an updated router according to one or more embodiments of the invention.

## DETAILED DESCRIPTION OF THE INVENTION

The invention is a method and apparatus for transparently caching DNS traffic. In the following description, numerous specific details are set forth to provide a more thorough description of embodiments of the invention. It is apparent, however, to one skilled in the art, that the invention may be practiced without these specific details. In other instances, well known features have not been described in detail so as not to obscure the invention.

### Embodiment of Computer Execution Environment (Hardware)

An embodiment of the invention can be implemented as computer software in the form of computer readable code executed on a general purpose computer such as computer 300 illustrated in Figure 3, or in the form of bytecode class files running on such a computer. A keyboard 310 and mouse 311 are coupled to a bi-directional system bus 318. The keyboard and mouse are for introducing user input to the computer system and communicating that user input to processor 313. Other suitable input devices may be used in addition to, or in place of, the mouse 311 and keyboard 310. I/O (input/output) unit 319 coupled to bi-directional system bus 318 represents such I/O elements as a printer, A/V (audio/video) I/O, etc.

Computer 300 includes a video memory 314, main memory 315 and mass storage 312, all coupled to bi-directional system bus 318 along with keyboard 310, mouse 311 and processor 313. The mass storage 312 may include both fixed and removable media, such as magnetic, optical or magnetic optical storage systems or any other available mass storage

technology. Bus 318 may contain, for example, thirty-two address lines for addressing video memory 314 or main memory 315. The system bus 318 also includes, for example, a 32-bit data bus for transferring data between and among the components, such as processor 313, main memory 315, video memory 314 and mass storage 312. Alternatively, multiplex data/address lines may be used instead of separate data and address lines.

In one embodiment of the invention, the processor 313 is a microprocessor manufactured by Motorola, such as the 680X0 processor or a microprocessor manufactured by Intel, such as the 80X86, or Pentium processor, or a SPARC microprocessor from Sun Microsystems, Inc. However, any other suitable microprocessor or microcomputer may be utilized. Main memory 315 is comprised of dynamic random access memory (DRAM). Video memory 314 is a dual-ported video random access memory. One port of the video memory 314 is coupled to video amplifier 316. The video amplifier 316 is used to drive the cathode ray tube (CRT) raster monitor 317. Video amplifier 316 is well known in the art and may be implemented by any suitable apparatus. This circuitry converts pixel data stored in video memory 314 to a raster signal suitable for use by monitor 317. Monitor 317 is a type of monitor suitable for displaying graphic images.

Computer 300 may also include a communication interface 320 coupled to bus 318. Communication interface 320 provides a two-way data communication coupling via a network link 321 to a local network 322. For example, if communication interface 320 is an integrated services digital network (ISDN) card or a modem, communication interface 320 provides a data communication connection to the corresponding type of telephone line, which comprises part of network link 321. If communication interface 320 is

a local area network (LAN) card, communication interface 320 provides a data communication connection via network link 321 to a compatible LAN. Wireless links are also possible. In any such implementation, communication interface 320 sends and receives electrical, electromagnetic or optical signals which carry digital data streams representing various types of information.

Network link 321 typically provides data communication through one or more networks to other data devices. For example, network link 321 may provide a connection through local network 322 to local server computer 323 or to data equipment operated by an Internet Service Provider (ISP) 324. ISP 324 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 325. Local network 322 and Internet 325 both use electrical, electromagnetic or optical signals which carry digital data streams. The signals through the various networks and the signals on network link 321 and through communication interface 320, which carry the digital data to and from computer 300, are exemplary forms of carrier waves transporting the information.

Computer 300 can send messages and receive data, including program code, through the network(s), network link 321, and communication interface 320. In the Internet example, remote server computer 326 might transmit a requested code for an application program through Internet 325, ISP 324, local network 322 and communication interface 320.

The received code may be executed by processor 313 as it is received, and/or stored in mass storage 312, or other non-volatile storage for later

execution. In this manner, computer 300 may obtain application code in the form of a carrier wave.

Application code may be embodied in any form of computer program product. A computer program product comprises a medium configured to store or transport computer readable code, or in which computer readable code may be embedded. Some examples of computer program products are CD-ROM disks, ROM cards, floppy disks, magnetic tapes, computer hard drives, servers on a network, and carrier waves.

The computer systems described above are for purposes of example only. An embodiment of the invention may be implemented in any type of computer system or programming or processing environment.

Embodiment of Software Apparatus for Transparently Caching DNS Traffic

One or more embodiments of the invention may be described by examining the layered model of networking and the peer relationships between the different layers. At the network layer, a peer relationship exists between each router that is connected by some type of wire. At the higher application layer, DNS entities (e.g., DNS resolvers and the local name servers) have a peer relationship with multiple hops in between (e.g., the routers). The routers at the network layer (the hops of the network layer) do not examine the information from application layer protocols. The routers merely transparently transfer the information between DNS clients and DNS servers.

In one or more embodiments of the invention, the layering model of networks is violated. DNS traffic is communicated from one machine to another machine through the use of name service ports. DNS traffic commonly arrives from and is transmitted to a specific DNS port (e.g., port 53). Consequently, based on the port information that is present in all IP packets, the routers have the ability to identify when DNS traffic is being transmitted versus when web or other traffic is being transmitted.

When an intermediate router (or hop in the network protocol layer) identifies that DNS information is in the packet it is transmitting across the internet, the routers violate the layering model and examine the information in the packet as if the router were a member of the application protocol. The information is then parsed and cached. Thus, the routers snoop on the DNS replies from a name-server and cache the intercepted replies. The routers also intercept DNS requests, and determine if the request can be served from the cache. If the cache contains the requested information, the router provides the response to the DNS query. If the cache does not contain the requested information, the router forwards the request to the next router or hop along the path to the name server.

Referring to the prior art system of Figure 1, at step 106, the resolver forwards the request to the local name server, and at step 108, the name server of the lowest level domain name is contacted. In one or more embodiments of the invention, the forwarding step 106 and the contacting step 108 are processed through routers that may intercept the transmissions. The routers examine the packet of information from the intercepted transmissions and store any necessary information in cache. Further, when the information is obtained from the name server and transmitted back to

the local name server at step 110, in one or more embodiments of the invention, the routers again intercept the transmission, parse the information, and cache the address information as it passes by on its way to the local name server.

Figure 5 demonstrates the process performed by an updated router according to one or more embodiments of the invention. The process starts at step 500. At step 502, the router examines the port information to determine if the current information is DNS traffic or some other type of traffic (e.g., web traffic). If the information is not DNS traffic, the router merely performs as normal and forwards the request to the next hop to its destination at step 512.

If the information is DNS traffic, the router parses the information at step 504. At step 506, the router determines if the parsed information (e.g., the requested address information) is in its cache. If the information is not in its cache, the router stores the relevant information (if any) in its cache at step 510 and forwards the request to the next hop in the information's path at step 512. If the information is in the router's cache, the router returns the requested information to the requestor at step 508. In this manner, the updated routers maintain their own cache and are capable of processing DNS translation requests.

Alternatively, between step 502 and step 504, if the information is DNS traffic, the router will automatically forward the DNS information to a preconfigured host. Routers are currently configured to recognize types of internet traffic and forward specified types of internet traffic to a specific location or host. Once the host receives the information, the host executes

the remaining steps 504-514. For example, the host parses the information at step 504 and searches its own cache for valid information at step 506. If there is any information to store in the cache (i.e., the DNS information is being returned), the information is stored in the host's cache at step 510. In such an embodiment, the router classifies and diverts packets to the configured host, and the host performs all additional functionality.

Referring to Figure 4, in one or more embodiments of the invention, one or more of the routers 210 may be modified as defined in Figure 4, to intercept, parse, and cache DNS information. For example, routers 404 and 406 may be updated. Consequently, when $Cl_1$ 212 requests a DNS translation from ns.syd.au 224, the request is forwarded through route 400 along routers 210 and updated routers 404 and 406. However, updated router 404 determines that it is DNS traffic, violates its network layer, and intercepts the request. Router 404 parses the requested information and determines if it is in its cache. If the requested information is in its cache, router 404 returns the result back to $Cl_1$ 212 (along route 400). If the requested information is not in its cache, it merely forwards the request to the next hop in pathway 400. Router 406, upon determining that the transmission is DNS traffic, intercepts the request and searches its cache. Upon determining that the relevant information is not in its cache, router 406 forwards the request to the next hop in pathway 400. The request is forwarded until it reaches the local name server ns.syd.au 224. Alternatively, as described above, in one or more embodiments, the router forwards the request (if it is DNS traffic) to a configured host that maintains the cache and processing capabilities.

The request is processed by ns.syd.au 224 and returned back to $Cl_1$ 212 along path 400. When the information reaches router 406 on its way back to

$Cl_1$ 212, router 406 intercepts the request, the router or configured host parses the address information, and stores the address information in cache. Router 406 then forwards the results to the next hop along path 400. Each updated router or configured host along path 400 will store the result in its cache.

Subsequent to the above request, if $Cl_4$ requests a similar DNS translation, the request would be forwarded along route 402. However, router 406 would identify the request as DNS traffic, router 406 intercepts the request, router 406 or a configured host parses the request, searches cache, and returns the requested information back to the previous hop on pathway 402. Consequently, the request by $Cl_4$ is serviced locally at router 406 or the configured host and does not need to be serviced in Australia at ns.syd.au 224.

As described above, according to one or more embodiments of the invention, the updated routers perform additional processing from other routers. The processing by the routers as described above and illustrated in Figure 5, includes viewing a portion of the DNS traffic, parsing the information, maintaining a database for cache storage, and searching cache for the information.

Some DNS name servers return different answers for client requests for the same host name. Such a response may be based on load-balancing considerations (e.g., the attempt to balance network traffic across multiple servers), or it may be chosen to direct the clients to "nearby" hosts. Use of such schemes may be less effective with the transparent DNS caching according to one or more embodiments of the invention. Some schemes provide for strategic geographic placement of cacheable data (e.g., routers that may cache web traffic) in order to provide the information for the highest

number of clients possible. The geographical scheme described in pending patent application number 09/081,860 entitled "Method and Apparatus for Effective Traffic Localization Through Domain Name System" which is hereby incorporated by reference, works well when used to determine which network routers are to be updated in accordance with one or more embodiments of the invention. In such a geographic scheme, the information returned is deliberately provided to be applicable to a large number of (if not all) DNS clients, with client-side computation to still achieve the load-balancing and traffic localization goals desired. Such a scenario reduces the network load as well as the latency observed in DNS translations.

Thus, a method and apparatus for encoding content characteristics for the retrieval of information is described in conjunction with one or more specific embodiments. The invention is defined by the claims and their full scope of equivalents.

## CLAIMS

1. A method for transparently processing DNS traffic comprising:

transmitting a request for information to a network router;

parsing said transmitted request;

searching cache for said requested information; and

returning said requested information if said requested information is in said cache.

2. The method of claim 1 further comprising:

forwarding said request to a next hop of said request if said requested information is not in said cache;

receiving said requested information;

parsing said requested information;

storing said requested information in said cache; and

forwarding said requested information to a next hop of said requested information.

3. The method of claim 1 wherein said information is internet protocol address information.

4. The method of claim 1 wherein said network router is applicable to one or more DNS clients based on geographical placement.

5. The method of claim 2 wherein said receiving step comprises transmitting said requested information from a name server.

6. A system comprising

a processor;

a memory coupled to said processor;

code executed by said processor configured to transparently process DNS traffic;

said code comprising:

a method transmitting a request for information to a network router;

a method parsing said transmitted request;

a method searching cache for said requested information; and

a method returning said requested information if said requested information is in said cache.

7. The system of claim 6 wherein said code further comprises:

a method forwarding said request to a next hop of said request if said requested information is not in said cache;

a method receiving said requested information;

a method parsing said requested information;

a method storing said requested information in said cache; and

a method forwarding said requested information to a next hop of said requested information.

8. The system of claim 6 wherein said information is internet protocol address information.

9. The system of claim 6 wherein said network router is applicable to one or more DNS clients based on geographical placement.

10. The system of claim 7 wherein said code for a method receiving said requested information comprises a method transmitting said requested information from a name server.

11. A computer program product comprising

a computer usable medium having computer readable program code embodied therein configured to transparently process DNS traffic, said computer program product comprising:

computer readable code configured to cause a computer to transmit a request for information to a network router;

computer readable code configured to cause a computer to parse said transmitted request;

computer readable code configured to cause a computer to search cache for said requested information; and

computer readable code configured to cause a computer to return said requested information if said requested information is in said cache.

12. The computer program product of claim 11 further comprising:

computer readable code configured to cause a computer to forward said request to a next hop of said request if said requested information is not in said cache;

computer readable code configured to cause a computer to receive said requested information;

computer readable code configured to cause a computer to parse said requested information;

computer readable code configured to cause a computer to store said requested information in said cache; and

computer readable code configured to cause a computer to forward said requested information to a next hop of said requested information.

13. The computer program product of claim 11 wherein said information is internet protocol address information.

14. The computer program product of claim 11 wherein said network router is applicable to one or more DNS clients based on geographical placement.

15. The computer program product of claim 12 wherein said computer readable code configured to cause a computer to receive comprises computer readable code configured to cause a computer to transmit said requested information from a name server.

16. The method of claim 1 wherein said cache is maintained by said network router.

17. The method of claim 1 wherein said cache is maintained by a configured host.

18. The system of claim 6 wherein said cache is maintained by said network router.

19. The system of claim 6 wherein said cache is maintained by a configured host.

20. The computer program product of claim 11 wherein said cache is maintained by said network router.

21. The computer program product of claim 11 wherein said cache is maintained by a configured host.

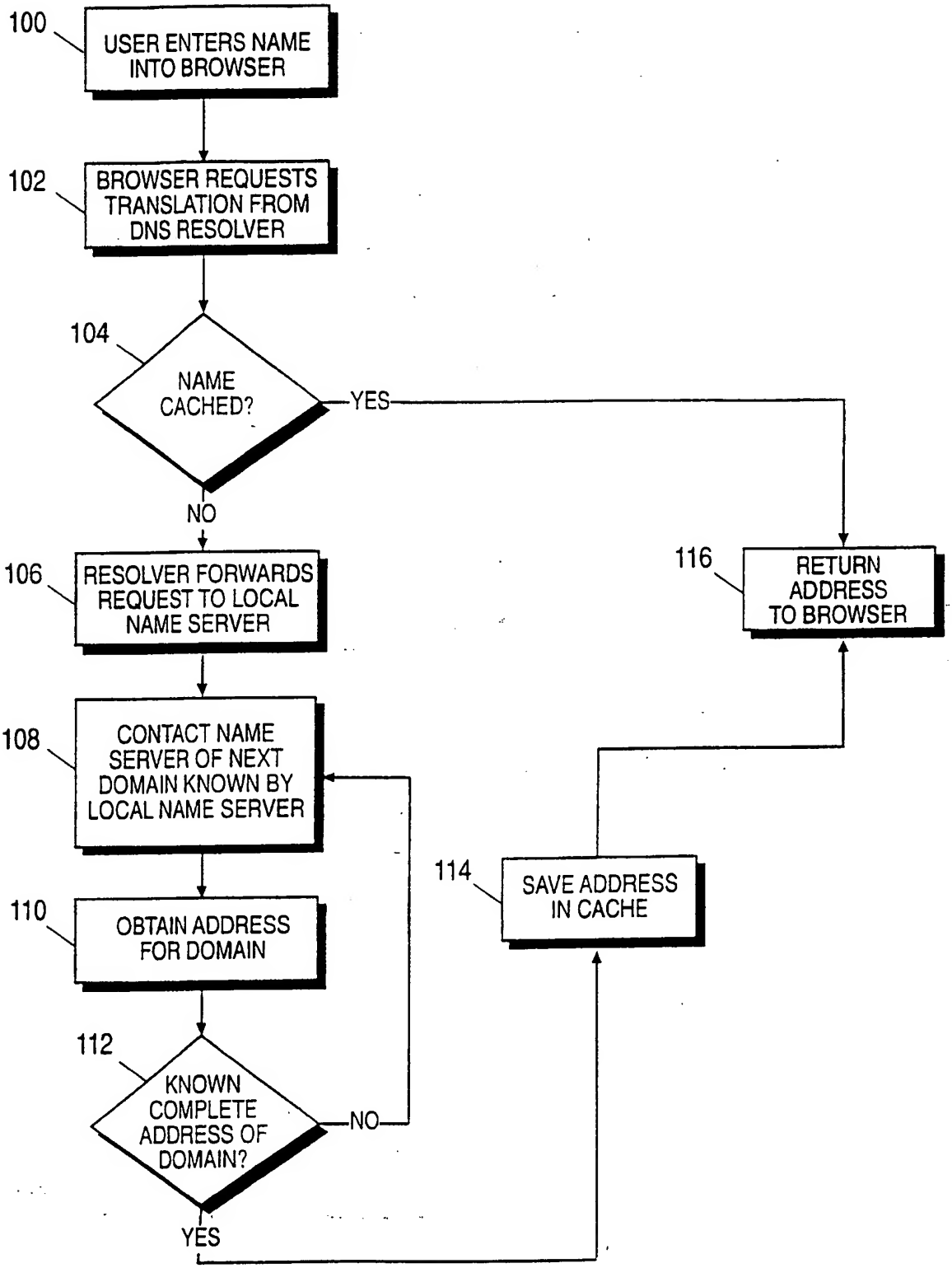

100
USER ENTERS NAME INTO BROWSER
102
BROWSER REQUESTS TRANSLATION FROM DNS RESOLVER
104
NAME CACHED?
YES
NO
106
RESOLVER FORWARDS REQUEST TO LOCAL NAME SERVER
108
CONTACT NAME SERVER OF NEXT DOMAIN KNOWN BY LOCAL NAME SERVER
110
OBTAIN ADDRESS FOR DOMAIN
112
KNOWN COMPLETE ADDRESS OF DOMAIN?
NO
YES
114
SAVE ADDRESS IN CACHE
116
RETURN ADDRESS TO BROWSER

FIG. 1

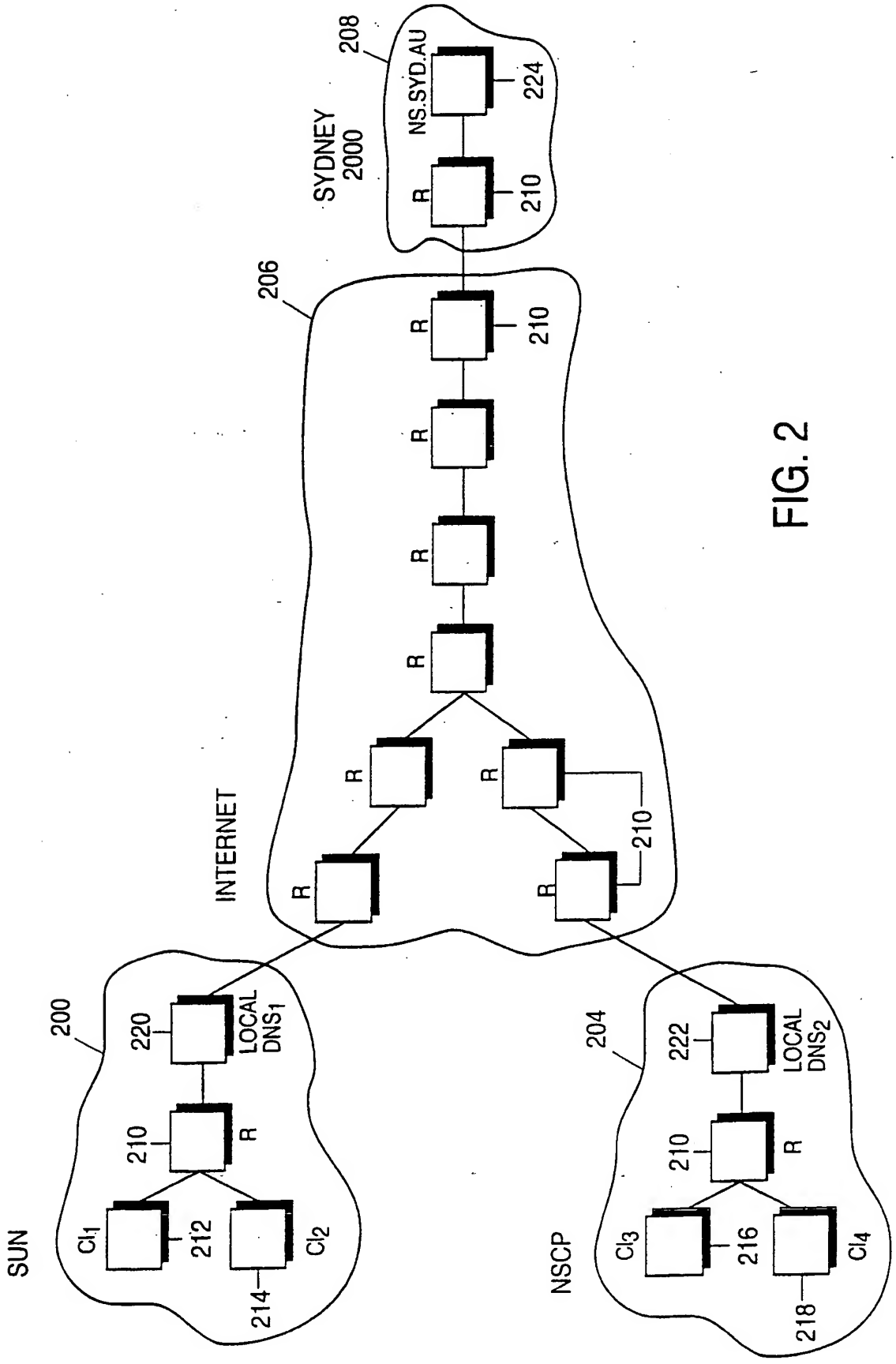

SYDNEY 2000
208
NS.SYD.AU
224
R
210
206
INTERNET
R
210
200
SUN
220
LOCAL DNS1
210
R
Cl1
212
Cl2
214
204
NSCP
222
LOCAL DNS2
210
R
Cl3
216
Cl4
218

FIG. 2

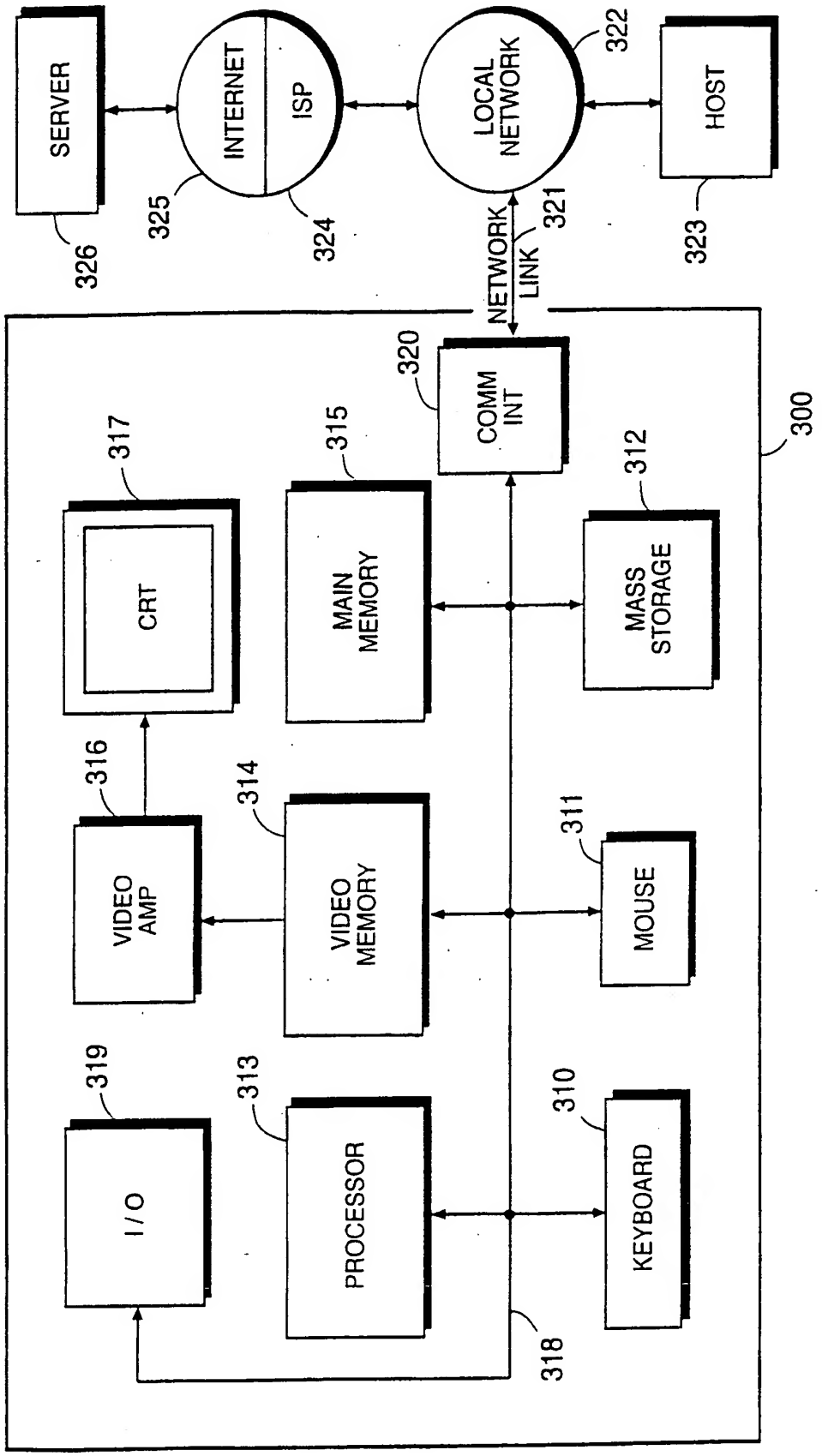

SERVER
INTERNET
ISP
LOCAL NETWORK
HOST
326
325
324
322
323
NETWORK LINK
321
300
COMM INT
320
CRT
317
MAIN MEMORY
315
MASS STORAGE
312
VIDEO AMP
316
VIDEO MEMORY
314
MOUSE
311
I/O
319
PROCESSOR
313
KEYBOARD
310
318

FIG. 3

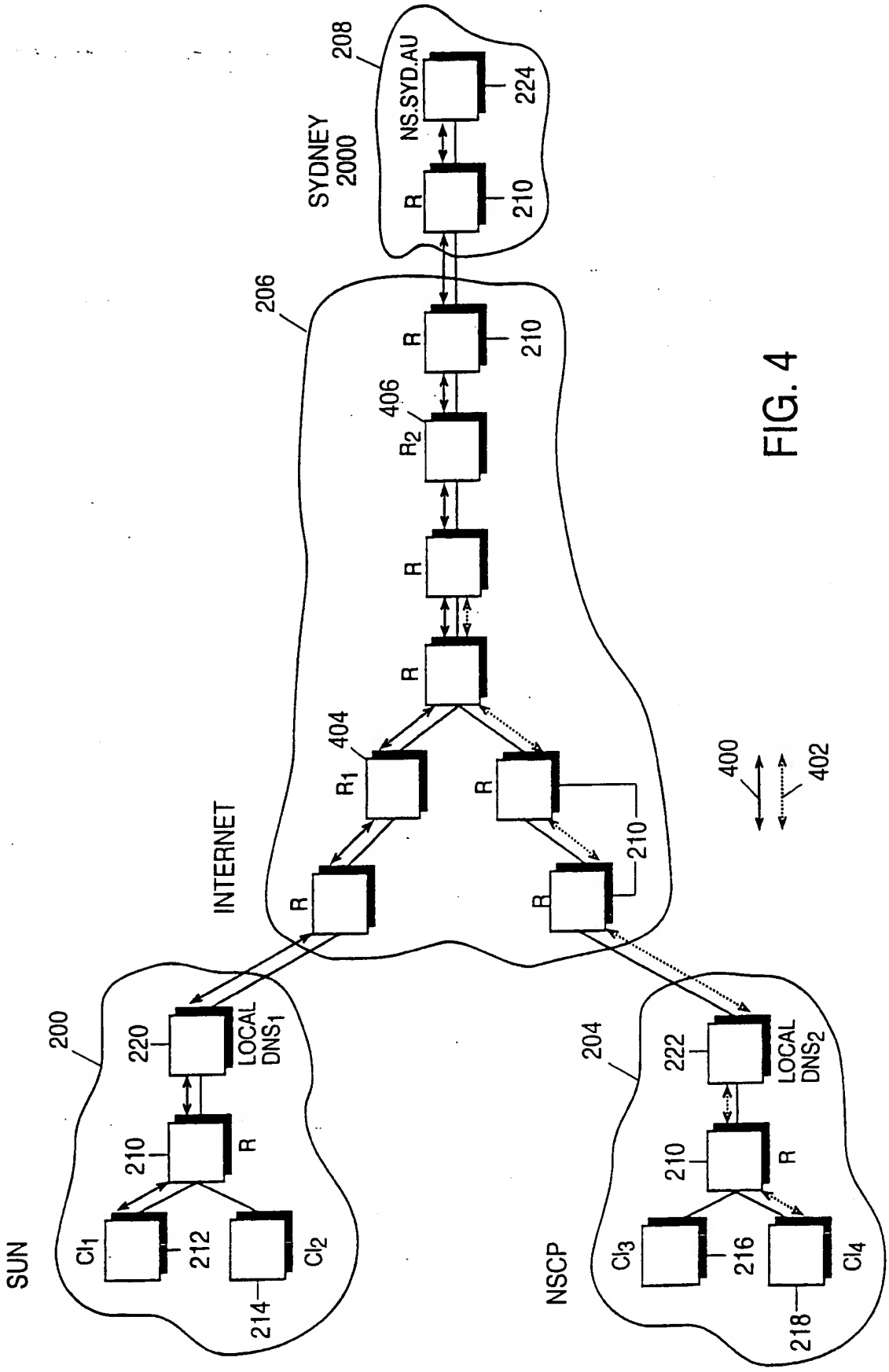

SYDNEY 2000
208
NS.SYD.AU
224
R
210
206
INTERNET
406
R2
R1
404
400
402
200
SUN
220
LOCAL DNS1
Cl1
212
Cl2
214
204
NSCP
222
LOCAL DNS2
Cl3
216
Cl4
218

FIG. 4

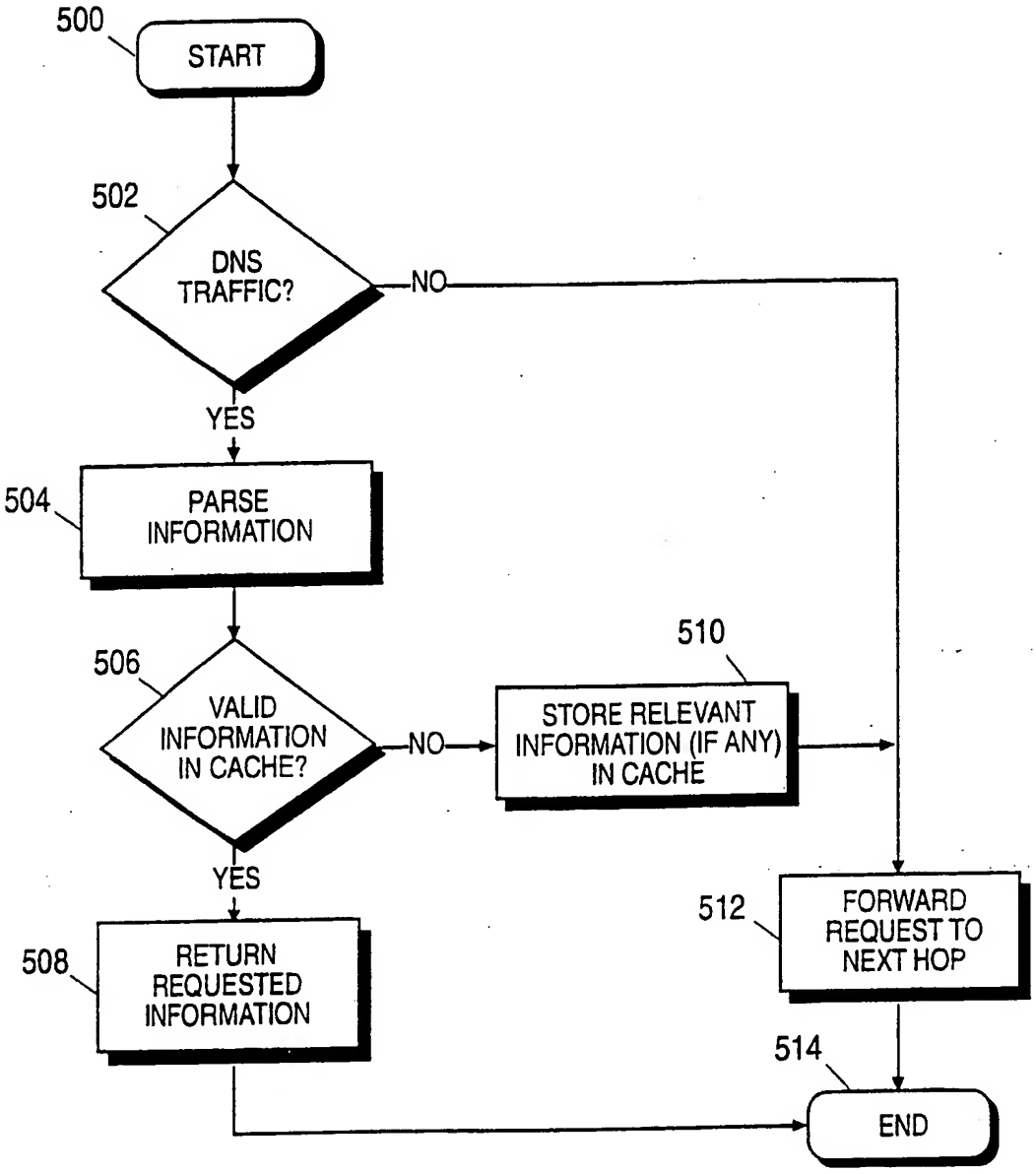

500
START
502
DNS TRAFFIC?
NO
YES
504
PARSE INFORMATION
506
VALID INFORMATION IN CACHE?
NO
510
STORE RELEVANT INFORMATION (IF ANY) IN CACHE
YES
508
RETURN REQUESTED INFORMATION
512
FORWARD REQUEST TO NEXT HOP
514
END

FIG. 5